

REMARKS

Applicant is in receipt of the Office Action mailed December 6, 2004. Claims 1-3, 6, 10-13, 15-21, 23, 25-30, 32-34, 36-37, 40-44, 46, 48-50, 53-54, 57-60, 62-63, and 66 have been amended. Claims 31, 45, and 61 have been canceled. New claims 67-71 have been added. Claims 1-30, 32-44, 46-60, and 62-71 are currently pending. Reconsideration of the present case is earnestly requested in light of the following remarks.

§102 Rejections

Claims 1-7 and 10-66 were rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Pat. No. 5,301,336 to Kodosky et al. (hereinafter “Kodosky”). Applicant respectfully traverses this rejection.

With respect to amended claim 1, Applicant submits that Kodosky does not teach or suggest several elements of the claim. For example, amended claim 1 recites in part, “displaying a first node for receiving user interface events in a block diagram for the graphical program”. Kodosky does not teach the concept of a node designed especially for receiving user interface events. Claim 1 has also been amended to further recite the element of, “receiving third user input specifying one or more user interface events to configure for the first node”. Kodosky does not teach or suggest the concept of a user providing input to specify one or more user interface events to configure for a node.

Kodosky relates generally to the field of graphical programming, in which a user creates a block diagram including a plurality of interconnected icons that represent a desired procedure. The Office Action makes reference to various passages in Kodosky that relate to nodes used in Kodosky’s graphical programs. However, none of the cited passages teach or even remotely suggest the features recited in amended claim 1.

For example, the Office Action cites Col. 15, lines 54-63 of Kodosky. This portion of Kodosky describes an iterative loop structure node. The loop structure node is operable to iteratively execute the icons placed within its borders in a loop. For example, FIG. 20f illustrates a virtual instrument icon placed inside the loop structure node. However, the loop structure node is not a node for receiving user interface events, as recited in claim 1. Kodosky makes no mention whatsoever of user interface events in

connection with the loop structure node. Furthermore, Kodosky certainly does not teach the element of, “receiving third user input specifying one or more user interface events to configure” for the loop structure node.

The Office Action also cites Col. 34, lines 27-41 of Kodosky. This portion of Kodosky relates generally to the order of execution of nodes in Kodosky’s diagrams. Kodosky’s diagrams execute according to a data flow model, and thus a node begins execution when all the input data for the node becomes available. Kodosky also describes the handling of a situation where a second node’s inputs are available before a first node finishes executing. Applicant respectfully submits that this portion of Kodosky bears little or no relevance to user interface events and certainly does not teach or suggest the elements recited in amended claim 1 of, “displaying a first node for receiving user interface events” and “receiving third user input specifying one or more user interface events to configure for the first node”.

The Office Action also cites Col. 38, lines 5-18 of Kodosky. This portion of Kodosky relates generally to execution states of structure nodes in Kodosky’s diagrams. For example, Kodosky describes how a node is scheduled to run by being placed on the run queue, how it is set to an active state when it is removed from the run queue and begins execution, etc. Again, Applicant respectfully submits that this portion of Kodosky bears little or no relevance to user interface events and certainly does not teach or suggest the elements recited in amended claim 1 of, “displaying a first node for receiving user interface events” and “receiving third user input specifying one or more user interface events to configure for the first node”.

The Office Action also cites Fig. 22 of Kodosky, stating that, “Figure 22 is a block diagram for the design example of Figure 21 and shows numerous event structure nodes represented by the virtual instrument icons such as ‘TEK5010FG’, ‘FLUKE 8840 A’, etc., as recited in column 17, line 48 – column 18, line 32.” However, what is actually stated at Col. 18 lines 7 – 17 is the following:

“Inside the iteration loop are two virtual instrument icons [i.e., the ‘TEK5010FG’, ‘FLUKE8840A’ icons]. The first takes as input an amplitude and a frequency and performs the appropriate IEEE-488 operations to set the function generator 208 of FIG. 21. The second performs the appropriate IEEE-488 operations to obtain a voltage measurement from the multimeter 210 of FIG. 21. The dotted line indicates that there is no data flow, but

ensures that they execute sequentially. These two icons represent simple virtual instruments that are easily designed using built-in high level IEEE-488 functions to communicate with the multimeter 210.”

Thus, Kodosky describes the ‘TEK5010FG’ and ‘FLUKE8840A’ icons as virtual instrument icons that take data values, such as an amplitude and frequency, as input and perform IEEE-488 operations to communicate with a multimeter. Therefore, neither of these icons constitutes a node for receiving user interface events as recited in amended claim 1. Furthermore, Kodosky does not teach receiving user input specifying one or more user interface events to configure for either of these icons. Applicant respectfully submits that these elements of amended claim 1 are simply not taught or suggested in Kodosky, either in the cited portions or anywhere else.

Amended claim 1 further recites the element of, “configuring the first node to receive the one or more user interface events specified by the third user input during execution of the graphical program”. Since Kodosky does not teach the element of “receiving third user input specifying one or more user interface events to configure for the first node”, Kodosky also does not teach “configuring the first node to receive the one or more user interface events specified by the third user input”.

The Office Action states that, “the user can interact with the block diagram by utilizing icons to build the block diagram; furthermore, as an example, the user interface event ‘CLEAR’ shown in Figure 25, can be used to remove certain wires from the block diagram.” The Office Action also cites various portions of Kodosky that relate to the user constructing a block diagram, such as Col. 9, lines 56-64.

Applicant agrees that Kodosky teaches the construction of a block diagram in response to user input, which is something that is common to many graphical programming systems. In fact, the Description of the Related Art in the present application references the Kodosky patent and states that, “The method disclosed in Kodosky et al allows a user to construct a diagram using a block diagram editor.” However, Kodosky in no way teaches or suggests the element recited in amended claim 1 of, “configuring the first node to receive the one or more user interface events specified by the third user input”. The graphical programming system taught in Kodosky simply does not allow a user to specify one or more user interface events to configure for a node

and is not operable to configure a node to receive one or more user interface events that have been specified by the user.

With respect to the statement in the Office Action that, “the user interface event [sic] ‘CLEAR’ shown in Figure 25, can be used to remove certain wires from the block diagram,” Applicant respectfully disagrees. Figure 25 illustrates a pull-down menu that includes a menu item labeled, ‘CLEAR’. A menu item is not at all the same as a user interface event, although Applicants notes in passing that the act of selecting a menu item may cause a user interface event to be generated. Applicant respectfully submits that the use of a menu item to remove wires from a block diagram in no way teaches or even remotely suggests the element recited in amended claim 1 of, “configuring the first node to receive the one or more user interface events specified by the third user input during execution of the graphical program”.

Thus, for at least the reasons provided above, Kodosky does not teach or suggest the elements recited in claim 1. Applicant therefore respectfully submits that independent claim 1, and claims dependent thereon, are patentable over Kodosky, and are thus allowable. Inasmuch as the other independent claims recite similar limitations as those of claim 1, Applicant submits that these claims, and claims respectively dependent thereon, are also allowable, for at least the reasons given above.

Applicant also submits that numerous ones of the dependent claims recite further distinctions over Kodosky. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time. Taking just one of the dependent claims as an example that recites further distinctions over Kodosky, new claim 67 recites the additional limitations of:

“displaying a list of user interface events;
wherein said receiving the third user input specifying the one or more user interface events to configure for the first node comprises receiving user input to select the one or more user interface events from the displayed list of user interface events.”

Applicant respectfully submits that Kodosky does not teach these limitations of claim 67.

§103 Rejections

Claims 8-9 were rejected under 35 U.S.C. 103(a) as being unpatentable over Kodosky and Zizzo (U.S. Pat. No. 6,578,174). Applicant respectfully traverses this rejection.

Applicant previously argued that there is no teaching, suggestion, or motivation to combine Kodosky and Zizzo in either of the references or in the prior art, and re-asserts this argument in the present amendment. As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolchem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis.

Furthermore, the showing of a suggestion, teaching, or motivation to combine prior teachings “must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not ‘evidence’.” *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.

Applicant respectfully submits that there is no clear teaching or suggestion for combining Kodosky and Zizzo and notes that the Office Action does not provide evidence of such a teaching or suggestion.

Furthermore, Applicant respectfully submits that it is nonobvious to combine Kodosky and Zizzo, and that even if Kodosky and Zizzo were combinable, which Applicant argues they are not, the resultant combination would still not produce the combination of elements recited in claims 8-9.

CONCLUSION

In light of the foregoing amendments and remarks, Applicant submits the application is now in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-58700/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,



Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 5/6/2005 JCH/JLB